Universidad de Castilla-la Mancha
Escuela Superior de Informática

Manual de instalación de un Nodo

Infraestructura distribuida para la construcción de paquetes Debian

Autor: José Luis Sanroma Tato

Índice general

1. Introducción **1**

1.1. Estructura del documento 1

2. ¿Qué es Node-Builder? **2**

2.1. ¿Por qué Node Builder? 3

3. Requisitos **4**

3.1. Requisitos de hardware 4

3.2. Paquetes necesarios 4

3.3. Requisitos de ejecución 5

3.4. El archivo preseed 5

4. Instalación **6**

4.1. Instalación de máquinas virtuales 6

5. Configuración **8**

5.1. Configuración de máquinas virtuales 8

5.2. Configuración de sudo 8

5.3. Configuración de repositorios 10

5.4. Configuración de Grub 10

Capítulo 1

Introducción

El presente documento describe los pasos necesarios para la instalación de un nodo en el sistema **distdeb**, un sistema distribuido para construcción de paquetes Debian.

Tanto el presente documento como el proyecto en sí está en constante desarrollo por lo tanto, el lector debe saber que puede haber cambios tanto en el manual como el código fuente.

La instalación de un Nodo dentro del sistema distdeb, se hará en el futuro mediante un paquete Debian.

1.1. Estructura del documento

Este manual se divide en 5 capítulos:

- Capítulo 2 ¿Qué es Node-Builder?
- Capítulo 3 Requisitos
- Capítulo 4 Instalación
- Capítulo 5 Configuración

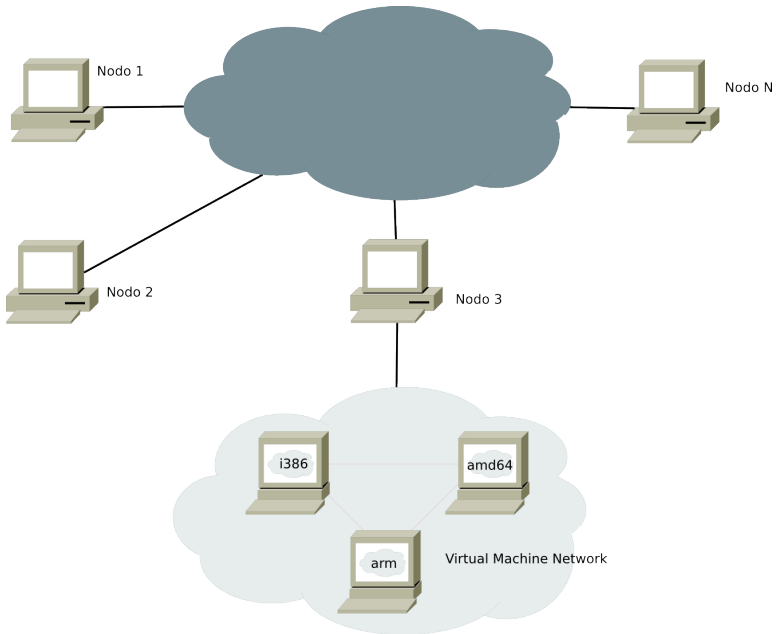
Cada uno de los capítulos describe las acciones necesarias y explicando con detalle todos los pasos a seguir para configurar un nodo que contiene a dos máquinas virtuales de arquitecturas *i386* y *amd64*

Capítulo 2

¿Qué es Node-Builder?

Node-Builder es la parte del sistema distribuido para contruir paquetes debian que se encarga de realizar la recontrucción de paquetes. Ayuda a distdeb y a los desarrolladores a generar los paquetes para las distintas arquitecturas para las cuales se quieren contruir los paquetes ya existentes y una vez construidos, subirlos al repositorio (Esta opción se implementará en posteriores versiones de desarrollo).

Node-Builder está formado por un computador que a su vez aloja una o varias máquinas virtuales donde se realiza la construcción de paquetes, comprobación de paquetes una vez construido, congelación de estado de máquina virtual y muchas más.



2.1. ¿Por qué Node Builder?

Mientras se desarrollaban las primeras iteraciones de **distdeb** y tras realizar algunas pruebas, se llegó a la conclusión de que era más ventajoso utilizar máquinas virtuales en lugar de *pbuilder*. Con las máquinas virtuales se tienen sistemas limpios donde realizar la construcción.

Al utilizar máquinas virtuales se tiene una opción que puede resolver el problema que se tiene cuando un paquete tarda mucho tiempo en construirse y el computador se tiene que apagar. Esto se soluciona con la utilización de "snapshots". Una "snapshot" o, traducido al español, una imagen, es la operación por la cual se toma un estado en un instante t cualquiera de la máquina virtual y se congela, pudiendo en cualquier momento ir a ese estado de "congelación" y, empezar o continuar desde ese estado la tarea que se estuviese desarrollando la proxima vez que se encienda el ordenador o la máquina virtual. Esta operación también tiene la ventaja de que se puede revertir y volver a un estado anterior cuando se requiera.

Una de las ventajas que tiene utilizar este sistema es que se pueden generar paquetes utilizando la emulación de algunos procesadores y que permitiría construir paquetes para más arquitecturas que si se usase pbuilder. A priori no sería necesario, pero de esta forma se generaliza la solución al problema dando soporte a más arquitecturas. Otra ventaja es que la máquina emulada, puede resultar en ocasiones más rápida que la propia máquina real. Se puede utilizar de ejemplo el famoso *Raspberry Pi*, en el cual la construcción de un paquete tardaría mucho más que en una máquina virtual emulando ese procesador en un PC de escritorio.

Capítulo 3

Requisitos

3.1. Requisitos de hardware

- Conexión a internet de banda ancha.
- Al menos 2GB de RAM.

3.2. Paquetes necesarios

Para la instalación y configuración se necesitan los siguientes paquetes:

- libvirt
- virtinst
- virt-viewer
- virt-manager

Todos ellos hacen referencia a la API libvirt por la cual se pueden gestionar máquinas virtuales con comandos y de forma casi automática.

Es una buena idea que se añada el usuario con el que se van a ejecutar los comandos al grupo libvirt, para no tener que recurrir a utilizar privilegios de administrador.

```
1    # adduser usuario libvirt
2    # adduser usuarios kvm
```

3.3. Requisitos de ejecución

Dada la naturaleza de libvirt, es necesario añadir lo siguiente a tu .bashrc para no tener que ejecutarlo continuamente cada vez que se inicia el computador

```
export LIBVIRT\_DEFAULT\_URI=qemu:///system
```

Se necesita crear una red[1] para interactuar con las máquinas virtuales y un directorio[2] donde se alojarán las máquinas virtuales. Este directorio debe existir en el sistema donde se realice la instalación. Además deberás introducir la ruta **absoluta** en el archivo src/xml/pool-distdeb.xml entre las etiquetas `<path>`, tal y como se muestra en el siguiente ejemplo:

```
<pool type="dir">
  <name>distdeb</name>
  <target>
    <path>/home/usuario/virtualMachines</path>
  </target>
</pool>
```

Ejecuta ahora el script *create_stuff.sh* para que se configuren tanto la red como el directorio donde se instalarán las máquinas virtuales.

3.4. El archivo preseed

Preseed[3] ofrece un mecanismo para responder a las preguntas que se realizan durante la instalación sin que tener que introducir las respuestas manualmente mientras se ejecuta la instalación. Gracias a esto es posible automatizar completamente la mayoría de instalaciones e incluso algunas características no están disponibles durante la instalación normal, como puede ser la instalación de algunos paquetes o la configuración de sudo por defecto.

El fichero pressed.cfg ya ha sido configurado de forma óptima para la instalación del sistema.

[1] Formato XML para redes con libvirt
[2] Formato XML para almacenamiento con libvirt
[3] Más información en su página web

Capítulo 4

Instalación

4.1. Instalación de máquinas virtuales

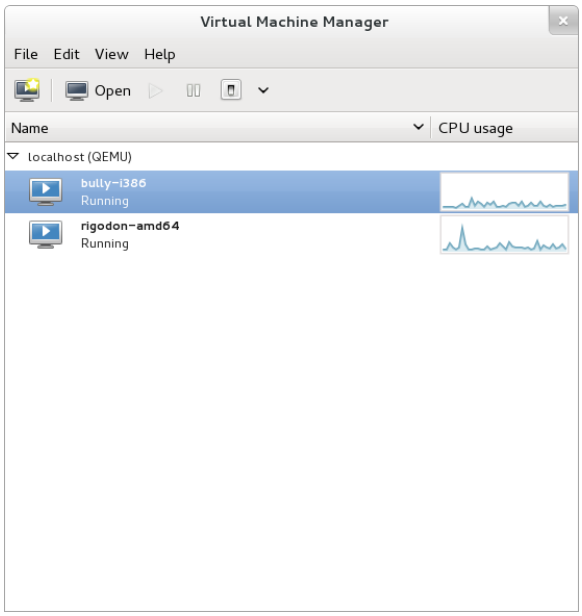
Para proceder a instalar las máquinas virtuales se ha creado un script que automatiza esta tarea, está en src/install_vm.sh. Este script genera automáticamente una instalación mínima del sistema Operativo Debian GNU/Linux rama testing de arquitectura i386 y amd64.

Si solamente se desa instalar **una** de las dos arquitecturas (i386 o amd64, basta con pasarla como parámetro al script y automáticamente la generará:

```
sh install_vm.sh <arquitectura>
```

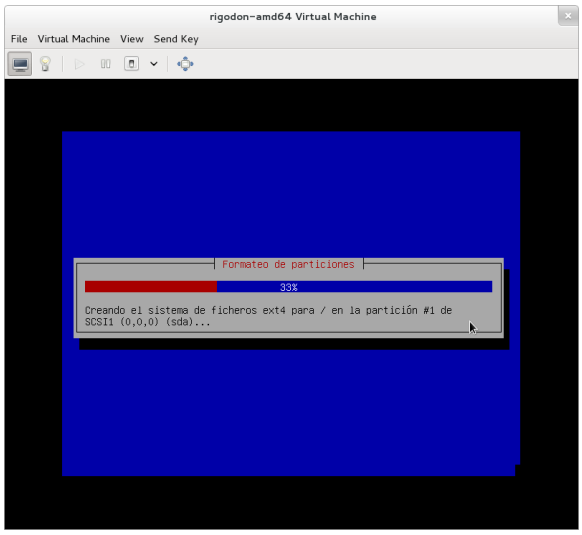
¡Importante! Si se detectase algún error durante la instalación es recomendable ejecutar la instalación por separado de ambas máquinas virtuales.

Al ejecutar el script, se realiza una descarga de internet y luego aparece la ventana del virt-manager donde se puede seguir el proceso de la instalación. Esta visión no se podrá ver mñas que unos gráficos con la carga de trabajo que tiene cada una de las máquinas virtuales, por lo que no aporta nada a simple vista.



Cabe la posibilidad de ver con más detalle el proceso, seleccionando una máquina virtual y pinchando sobre el botón “open” del menú de virt-manager.

Lo que se verá es la típica instalación de debian pero donde el usuario final no tiene que hacer nada porque todo ya se ha configurado automaticamente.


rigodon-amd64 Virtual Machine
File Virtual Machine View Send Key
Formateo de particiones
33%
Creando el sistema de ficheros ext4 para / en la partición #1 de SCSI1 (0,0,0) (sda)...

Cuando todo el proceso de instalar una o ambas máquinas virtuales ha finalizado, las máquinas se apagan y se puede continuar con su configuración.

Capítulo 5

Configuración

Esta es la sección referida a la configuración de las máquinas virtuales. Si solamente se ha instalado una máquina virtual, no tiene por qué utilizar *clusterssh*, pase directamente al siguiente punto, configuración de *sudo* 5.2.

5.1. Configuración de máquinas virtuales

Por defecto la mayoría de los paquetes están listos e instalados en las máquinas virtuales, incluso se tiene creado un usuario por defecto que puede utilizar el comando `sudo`. Este usuario tiene por nombre `arco` y por contraseña `arco`. Además, se necesita instalar algún paquete más en el computador anfitrión, ya que la conexión entre el anfitrión y las máquinas virtuales se realizad por *SSH*.

```
# aptitude install sshpass clusterssh
```

El paquete *clusterssh* permite ejecutar el mismo comando en n máquinas conectadas por *SSH*, viene muy bien para no tener que repetir tareas en ambas máquinas virtuales. Con el uso de *cssh* se hará todo de una sola vez. Se verá más adelante su uso en la subsección 5.2

5.2. Configuración de sudo

Si solamente tiene una máquina virtual no utilice *clusterssh*, puede utilizar *ssh* o meterse normalmente en la máquina virtual para realizar la configuración.

Una vez que *clusterssh* está instalado en el sistema anfitrión, lo que hay que hacer ahora es configurar las máquinas virtuales para que no soliciten la contraseña cada vez que se utilice la orden `sudo`. Por lo tanto, mediante el siguiente comando se procede a conectarse a las dos máquinas virtuales instaladas previamente.

```
cssh arco@IP1 arco@IP2
```

Donde `IP1` y `IP2` son las *IP* correspondientes a cada una de las máquinas virtuales, por defecto 192.168.122.11 y 192.168.122.12.

```
sudo nano /etc/sudoers
```

Y se añade al archivo para que no pida la contraseña cada que vez que se usa el comando `sudo` la línea **%arco ALL=NOPASSWD:ALL**:

```
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%arco   ALL=NOPASSWD:ALL
# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

5.3. Configuración de repositorios

Por defecto las instalación de Debian marca unos repos con el nombre en clave de la distribución testing (actualmente *wheezy* y proximamente *Jessie*).

Se procede a cambiar los repositorios para que apunten a "testing" y evitar así problemas con las nomenclaturas, de esta forma, las máquinas virtuales siempre serán rama "testing" de Debian.

```
sudo nano /etc/apt/sources.list
```

Y se sustituye *wheezy* por testing, quedando el archivo de la siguiente forma:

```
deb http://ftp.es.debian.org/debian/ testing main
deb-src http://ftp.es.debian.org/debian/ testing main

deb http://security.debian.org/ testing/updates main
deb-src http://security.debian.org/ testing/updates main
```

5.4. Configuración de Grub

Este paso es opcional y sirve para desactivar el tiempo de espera que tiene Grub para iniciar las máquinas virtuales, ya que "libvirt" no sabe a priori cuando una máquina virtual está lista para realizar las conexiones ssh.

Por ello, se edita el fichero */boot/grub/grub.cfg* cambiando el `tiemout` de 5 a 0.

Una vez se tiene todo esto, ya no es necesario realizar ninguna tarea más en la máquina virtual, ya que serán los scripts los que se encarguen de realizar todas las tareas.